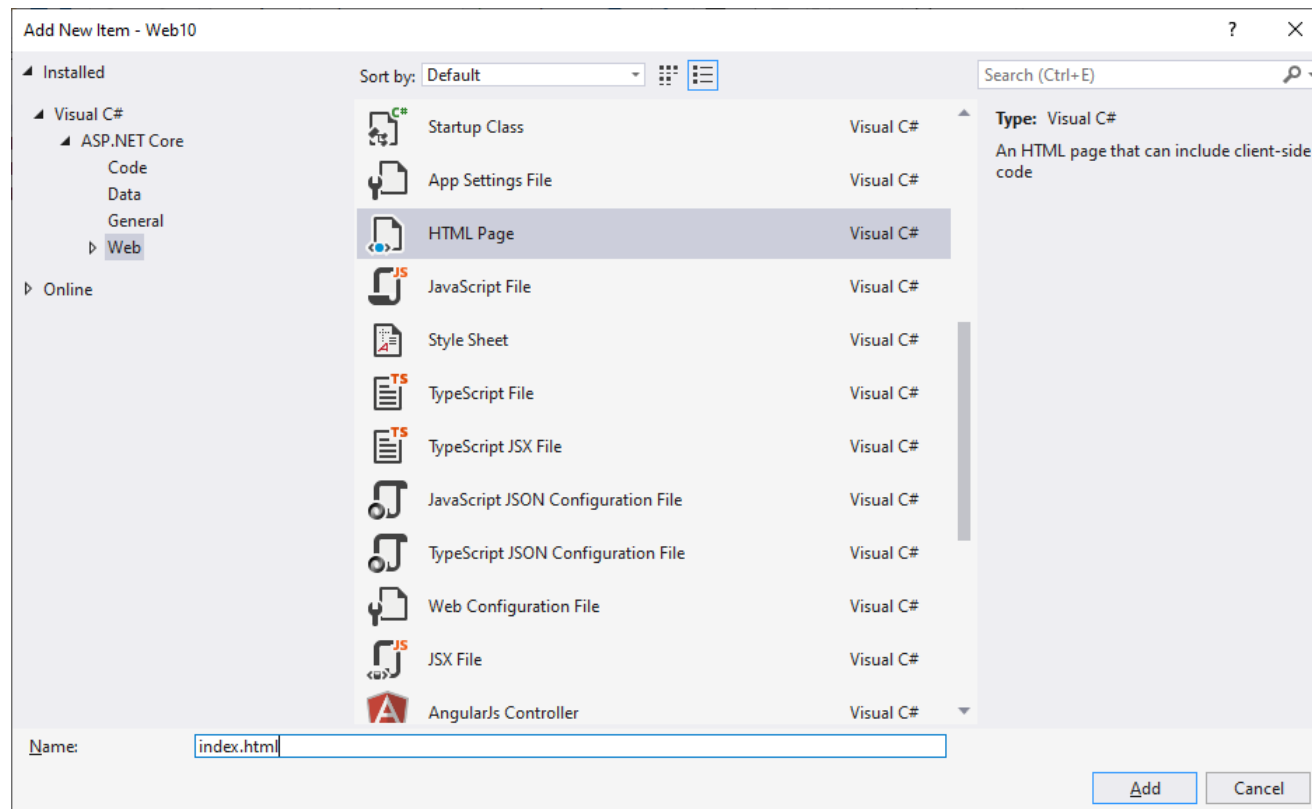


# Biblioteka za klijentsku validaciju

# Strana index.html

Unutar foldera wwwroot dodajte stranu index.html



# Postavljanje početne strane

```
var app = builder.Build();

// Configure the HTTP request pipeline.
if (!app.Environment.IsDevelopment())
{
    app.UseExceptionHandler("/Home/Error");
}
app.UseDefaultFiles();
app.UseStaticFiles();

app.UseRouting();

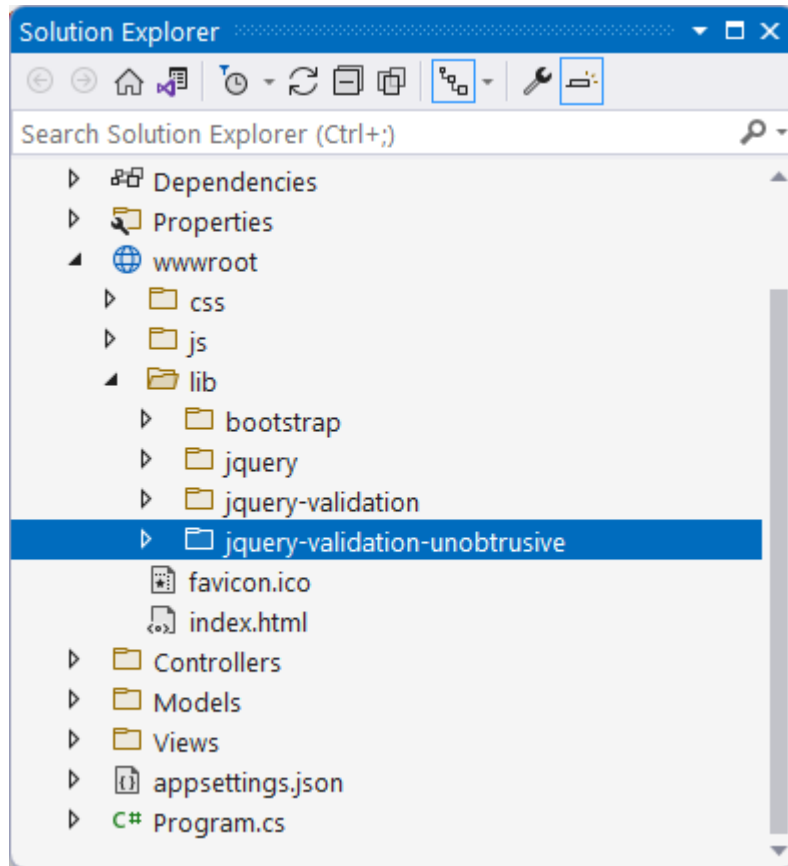
app.UseAuthorization();

app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Home}/{action=Index}/{id?}");

app.Run();
```

Strana index.html postaje početna strana web sajta

# Biblioteka jquery-validation-unobtrusive



# Biblioteka jQuery unobtrusive validation

- Microsoft je preradio jquery biblioteku za validaciju i od nje napravio biblioteku za tzv. unobtrusive(nenametljivu) validaciju
- Validacija ulaznih kontrola html forme vrši se korišćenjem atributa za validaciju a ne pisanjem klijentskog koda
- Da bi ova biblioteka funkcionisala neophodno je da se referencira jquery biblioteka i jquery biblioteka za validaciju
- Funkcionalnost unobtrusive biblioteke koriste tag helperi koji se koriste u cshtml fajlovima
- Ova biblioteka se kod ASP.NET Core MVC aplikacija referencira posredstvom parcijalnog pogleda `_ValidationScriptPartial.cshtml`

# Atributi za klijentsku validaciju

- **Atributi <input> elementa:**

- **data-val="true"** - omogućena klijentska validacija
- **data-val-length-min="3"** - minimalna dužina tekst polja
- **data-val-length-max="20"** - maksimalna dužina tekst polja
- **data-val-required="Unesite ime"** – poruka greške ako je polje prazno
- **data-val-length="Minimalno 3 a maksimalno 20 karaktera"** – poruka greške ako nije zadovoljena minimalna ili maksimalna dužina tekst polja

- **Atributi <span> elementa:**

- **data-valmsg-for** omogućava da se specificira name svojstvo kontrole za koju span element prikazuje validacionu grešku.
- **data-valmsg-replace** označava da će se sadržaj span elementa zameniti sadržajem validacione greške

# Referenciranje biblioteke za unobtrusive validaciju

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Unobtrusive validacija</title>
</head>
<body>

  <script src="lib/jquery/dist/jquery.min.js"></script>
  <script src="lib/jquery-validation/dist/jquery.validate.min.js"></script>
  <script src="lib/jquery-validation-unobtrusive/jquery.validate.unobtrusive.min.js"></script>
</body>
</html>
```

# Parcijalni pogled \_ValidationScriptPartial.cshtml



The image shows a code editor window titled "Web10 - \_ValidationScriptsPartial.cshtml". The editor contains two lines of HTML code, each enclosed in a script tag. The first line is `<script src="~/lib/jquery-validation/dist/jquery.validate.min.js"></script>` and the second line is `<script src="~/lib/jquery-validation-unobtrusive/jquery.validate.unobtrusive.min.js"></script>`. The code is color-coded: the opening and closing tags are in blue, the `src` attribute is in red, and the file paths are in green. The editor interface includes a tab at the top, a scrollbar on the right, and a status bar at the bottom showing "119%", "No issues found", and "Ln: 1 Ch: 1 SPC CRLF".

```
<script src="~/lib/jquery-validation/dist/jquery.validate.min.js"></script>  
<script src="~/lib/jquery-validation-unobtrusive/jquery.validate.unobtrusive.min.js"></script>
```



# Css klase koje koristi biblioteka

```
<head>
  <meta charset="utf-8" />
  <title>Unobtrusive validacija</title>
  <style>

    .field-validation-error {
      font-style: italic;
      font-size: 0.8em;
      color: red;
    }

    .field-validation-valid {
      display: none;
    }

    .validation-summary-valid {
      display: none
    }

    .validation-summary-errors {
      font-weight: bold;
      color: red;
    }

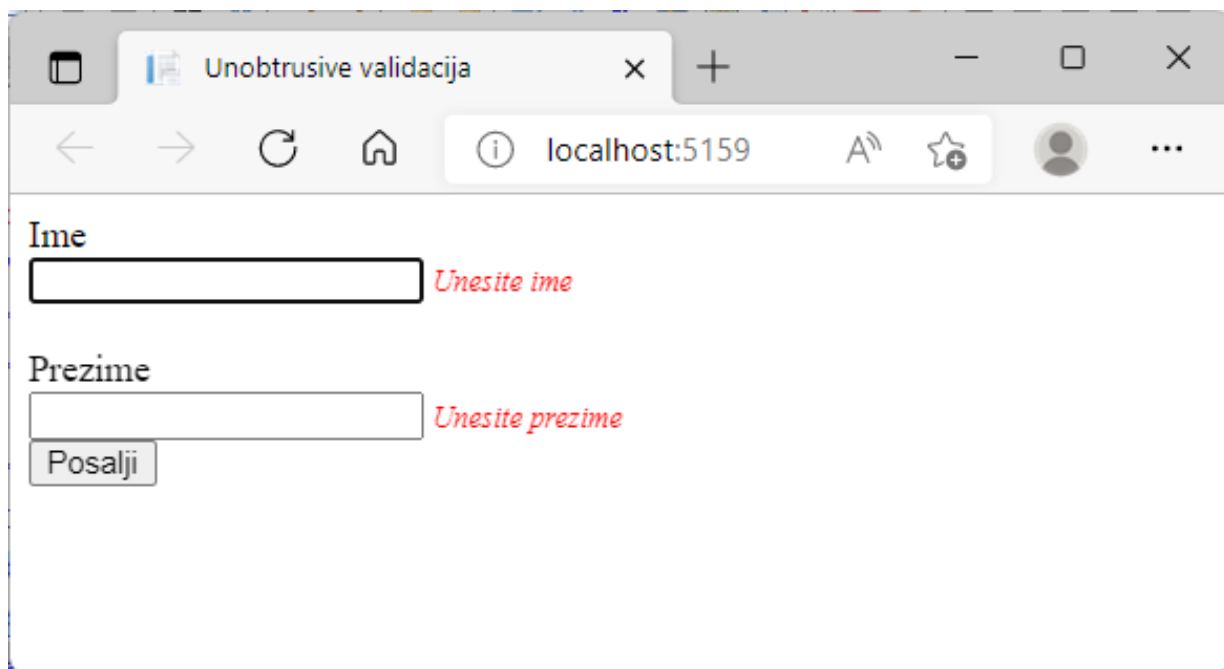
  </style>
</head>
```

# Primer validacije tekst polja

```
<label for="Ime">Ime</label>
<br>

<input data-val="true" data-val-length="Minimalno 3 a maksimalno 20 karaktera"
      data-val-length-min="3" data-val-length-max="20" data-val-required="Unesite ime"
      id="TextIme" name="Ime" type="text">
  <span data-valmsg-for="Ime" data-valmsg-replace="true"></span> <br>
<br>
```

# Prikaz validacionih grešaka



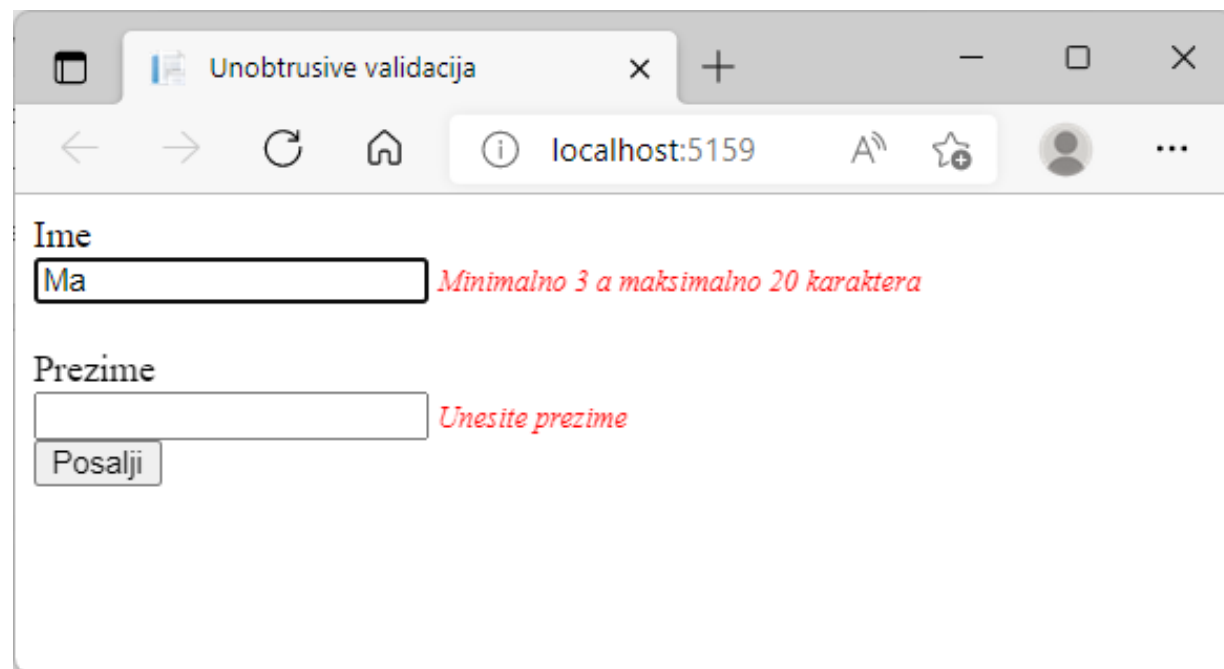
Unobtrusive validacija

localhost:5159

Ime  *Unesite ime*

Prezime  *Unesite prezime*

Posalji



Unobtrusive validacija

localhost:5159

Ime  *Minimalno 3 a maksimalno 20 karaktera*

Prezime  *Unesite prezime*

Posalji

# Elementi nakon validacije

```
<input data-val="true" data-val-length="Minimalno 3 a maksimalno 20 karaktera" data-val-length-min="3" data-val-length-max="20" data-val-required="Unesite ime" id="TextIme" name="Ime" type="text" class="input-validation-error" aria-describedby="TextIme-error" aria-invalid="true">
```

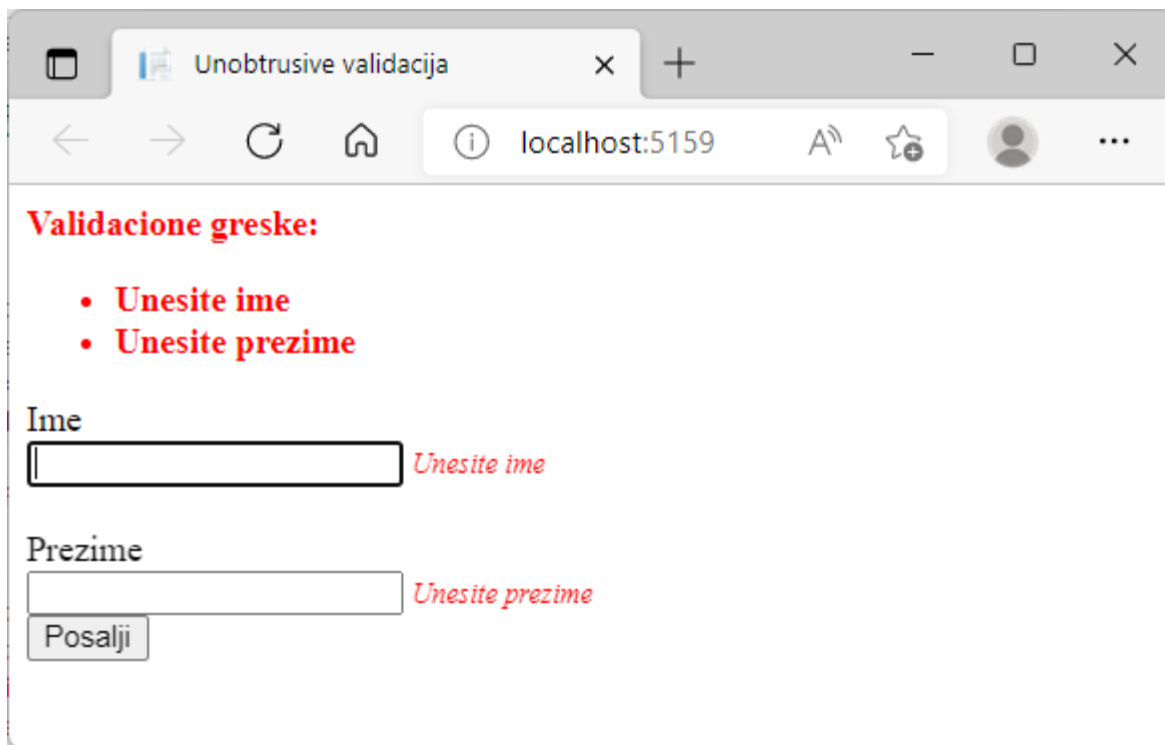
```
<span data-valmsg-for="Ime" data-valmsg-replace="true" class="field-validation-error">  
<span id="TextIme-error" class="">Minimalno 3 a maksimalno 20 karaktera</span>  
</span>
```

# Prikaz validacionih grešaka u vidu ul liste

```
<div data-valmsg-summary="true" class="validation-summary-valid">  
  <span>Validacione greske:</span>  
  <ul>  
  </ul>  
</div>
```

Atribut **data-valmsg-summary** div elementa označava da će se validacione greške ulaznih polja prikazati unutar unordered liste koja je deo ovog elementa

# Prikaz validacionih grešaka u vidu ul liste



The screenshot shows a web browser window with the title "Unobtrusive validacija" and the address bar displaying "localhost:5159". The page content includes a red heading "Validacione greske:" followed by a bulleted list of two error messages: "Unesite ime" and "Unesite prezime". Below the list, there are two input fields. The first is labeled "Ime" and has a red error message "Unesite ime" next to it. The second is labeled "Prezime" and has a red error message "Unesite prezime" next to it. At the bottom left, there is a button labeled "Posalji".

**Validacione greske:**

- Unesite ime
- Unesite prezime

Ime  *Unesite ime*

Prezime  *Unesite prezime*

Posalji

# Lista nakon validacije

```
<div data-valmsg-summary="true" class="validation-summary-errors">  
  <span>Validacione greske:</span>  
  <ul>  
    <li>Unesite ime</li>  
    <li>Unesite prezime</li>  
  </ul>  
</div>
```

# Pitanje 1

Klijentska biblioteka jQuery unobtrusive validation omogućava:

- a. klijentsku validaciju za ulazno polje forme pisanjem javascript koda
- b. klijentsku validaciju za ulazno polje forme korišćenjem atributa za validaciju
- c. serversku validaciju

Odgovor: b



# Pitanje 2

Dato je sledeće polje forme:

```
<input data-val="true" data-val-required="Unesite ime" name="Ime" type="text">.
```

Da bi span element prikazivao validacionu grešku za ulazno polje forme potrebno je unutar span elementa:

- a. Postaviti atribut `data-valmsg-for="Ime"` i atribut `data-valmsg-replace="true"`
- b. Postaviti atribut `data-valmsg-for="Ime"` i atribut `data-valmsg-replace="false"`
- c. Postaviti atribut `data-valmsg-for="true"` i atribut `data-valmsg-replace="Ime"`

Odgovor: a

# Pitanje 3

Da bi biblioteka za jquery unobtrusive validaciju prikazivala validacione greške unutar <ul> elementa koji je u sastavu nekog diva, potrebno je unutar tog div elementa:

- a. Postaviti atribut data-valmsg-replace="true"
- b. Postaviti atribut summary="true"
- c. Postaviti atribut data-valmsg-summary="true"

Odgovor: c

Održavanje stanja

# Održavanje stanja

- Kada korisnik pristupi web sajtu, web server šalje stranu koju je korisnik zahtevao
- Ako korisnik zatraži drugu stranu, web server je locira i vraća je nazad korisniku
- Sa stanovišta servera ova dva zahteva su potpuno nezavisna
- Ovo može biti problem ukoliko su na drugoj strani neophodni podaci uneti na prvoj strani
- Browseru je potreban mehanizam da čuva informacije koje predstavljaju stanje korisničke sesije
- Browser može slati te podatke kao deo svakog zahteva tako da server može detektovati korisnika

# Cookie

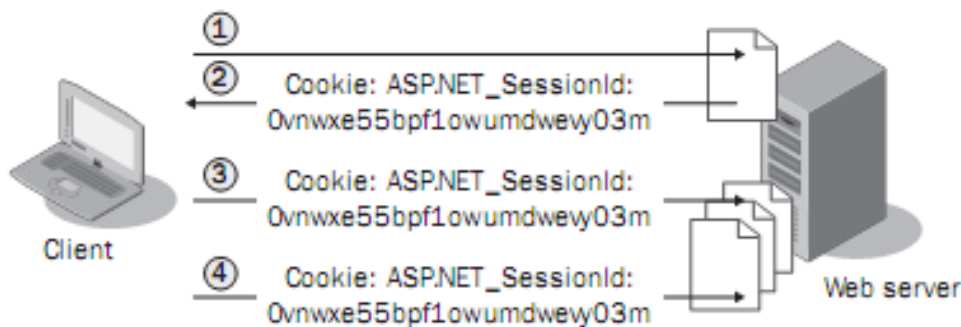
- Cookie je tekstualni fajl koji se čuva u folderu koji pridružen web browseru dok korisnik pregleda neki web sajt
- Svaki browser ima svoje kolačiće
- Najčešće korišćena primena kukija je identifikacija korisnika kada on posećuje više strana web aplikacije
- Postoje dve vrste cookie-ja
  - privremeni cookie – čije je vreme života ograničeno na korisničku sesiju
  - stalan ili perzistentan cookie – čije se vreme života definiše

# Cookie

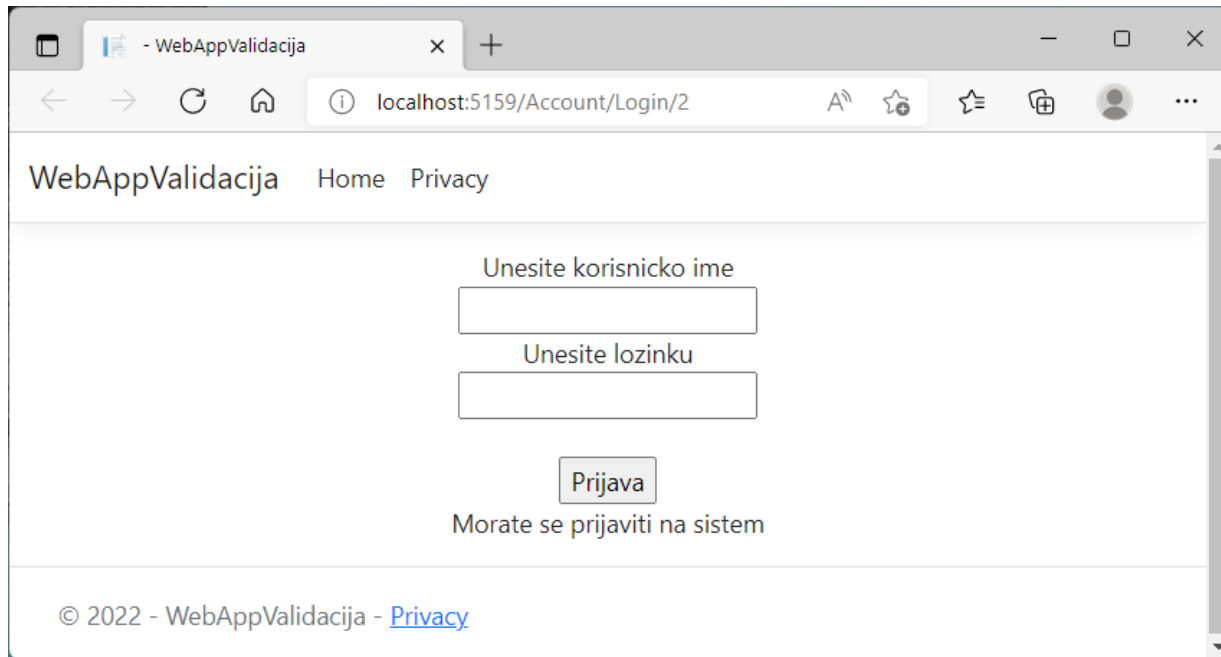
- Maksimalna veličina cookie-ja je 4Kb
- Kolačići su dizajnirani da se šalju ka web serveru i od web servera korisnika pri svakom zahtevu
- Podaci sačuvani unutar kolačića se dele između tabova browsera
- Kolačić se može kreirati i iščitavati pomoću klijentskog koda
- **HttpOnly** cookie se ne može čitati pomoću klijentskog koda i prvi put je implementiran 2002 godine i kreira se uz pomoć serverskog koda

# Praćenje klijenta uz korišćenje kukija

- U koraku 1 klijent zahteva stranu od web servera, i pošto ranije nije posećivao stranu ne poseduju kuki
- U koraku 2 web server pored odgovora šalje korisniku kuki
- Klijent šalje kuki sa svakim sledećim zahtevom čime biva identifikovan (koraci 3 i 4)



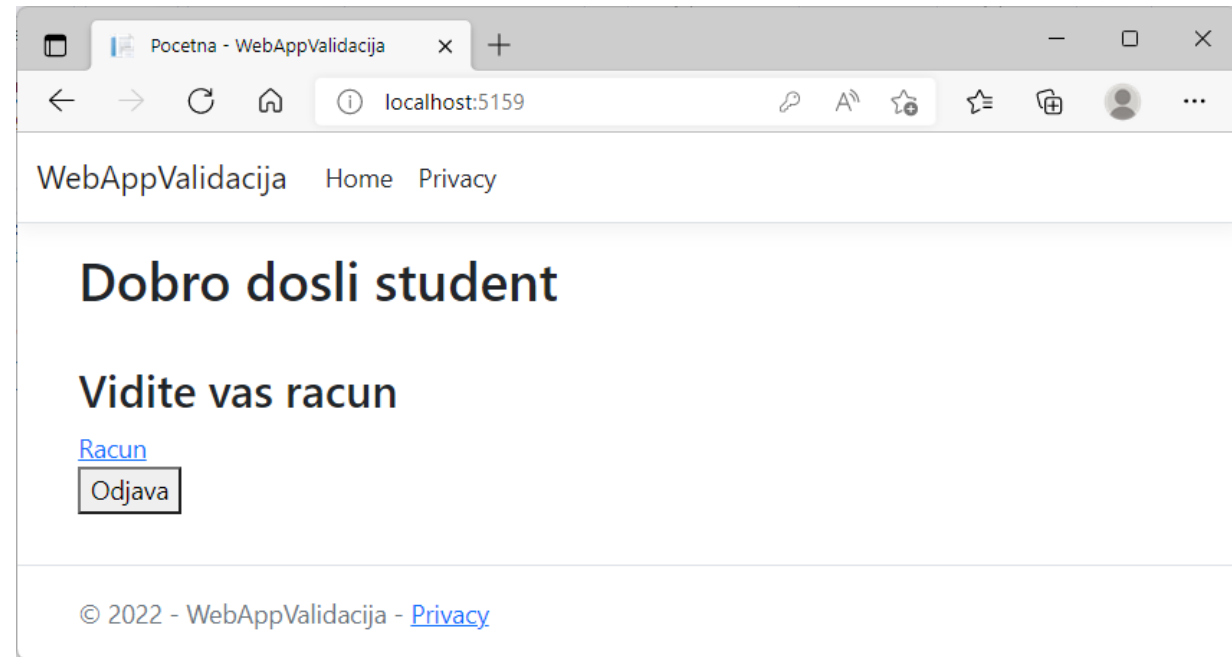
# Prijava na sistem



The screenshot shows a web browser window with the address bar at `localhost:5159/Account/Login/2`. The page title is "WebAppValidacija" and the navigation menu includes "Home" and "Privacy". The main content area contains a login form with the following elements:

- Label: "Unesite korisnicko ime" above an input field.
- Label: "Unesite lozinku" above an input field.
- Button: "Prijava" (Login).
- Text: "Morate se prijaviti na sistem" (You must log in to the system).

The footer contains the text: "© 2022 - WebAppValidacija - [Privacy](#)".



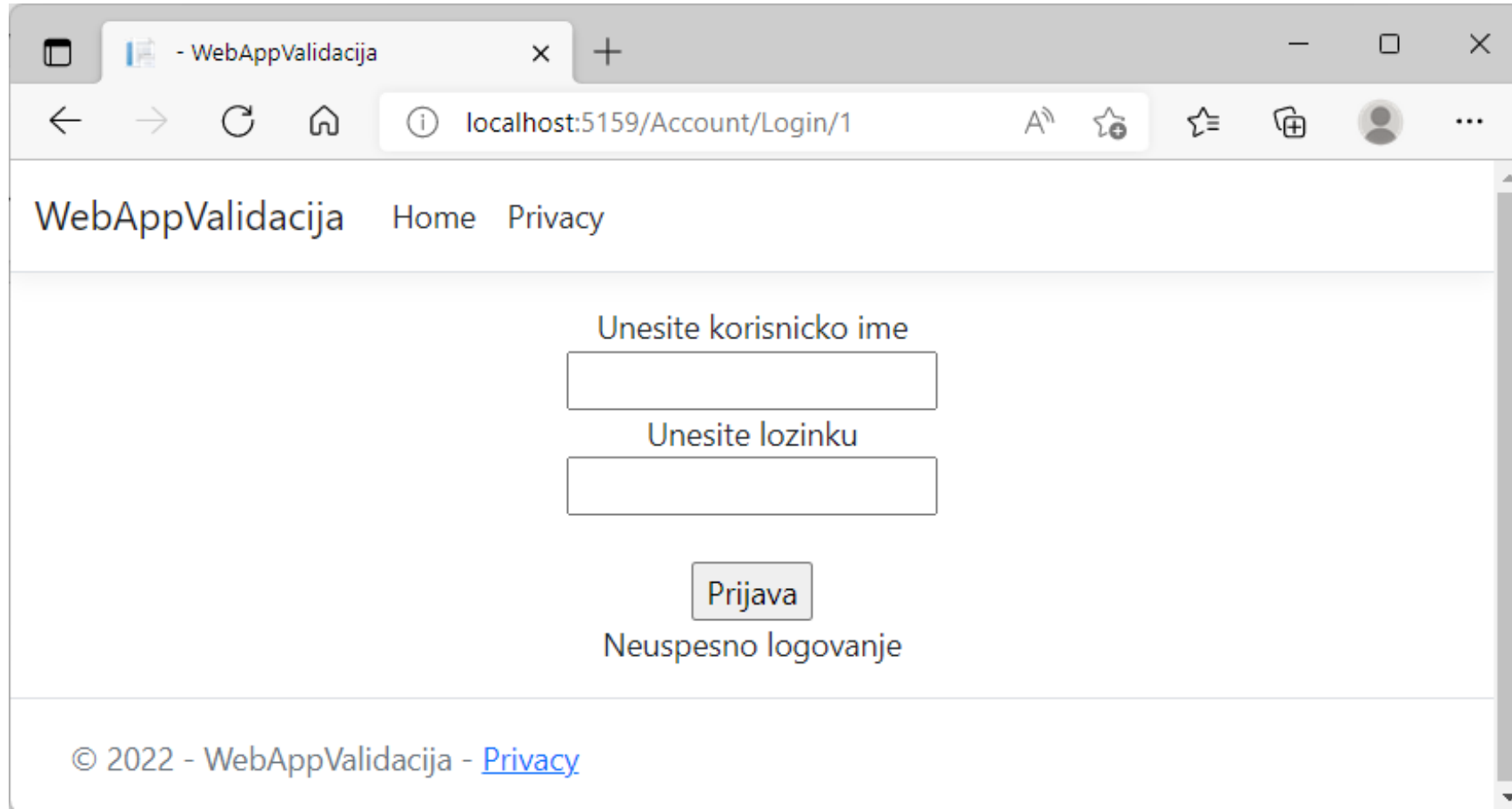
The screenshot shows a web browser window with the address bar at `localhost:5159`. The page title is "Pocetna - WebAppValidacija" and the navigation menu includes "Home" and "Privacy". The main content area displays a welcome message:

- Text: "Dobro dosli student" (Welcome student).
- Text: "Vidite vas racun" (View your account).
- Text: [Racun](#) (Account).
- Text: [Odjava](#) (Logout).

The footer contains the text: "© 2022 - WebAppValidacija - [Privacy](#)".



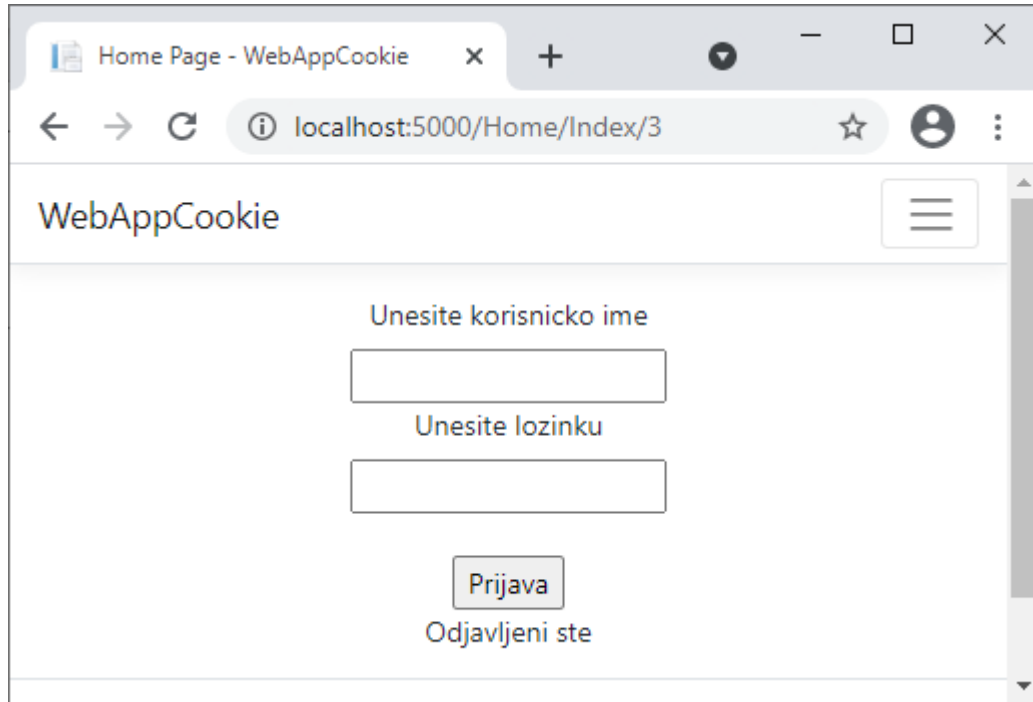
# Neuspesna prijava



The screenshot shows a web browser window with the following elements:

- Browser tab: - WebAppValidacija
- Address bar: localhost:5159/Account/Login/1
- Page title: WebAppValidacija
- Navigation links: Home, Privacy
- Form fields:
  - Unesite korisnicko ime (User name input field)
  - Unesite lozinku (Password input field)
- Submit button: Prijava
- Message: Neuspesno logovanje (Failed login)
- Footer: © 2022 - WebAppValidacija - [Privacy](#)

# Odjava



The screenshot shows a web browser window with the following elements:

- Browser tab: Home Page - WebAppCookie
- Address bar: localhost:5000/Home/Index/3
- Page title: WebAppCookie
- Form fields:
  - Label: Unesite korisnicko ime
  - Input field for username
  - Label: Unesite lozinku
  - Input field for password
- Buttons:
  - Prijava (Login)
  - Odjavljeni ste (You are logged out)

# Baza podataka LogovanjeMVC

```
CREATE DATABASE LogovanjeMVC
COLLATE Serbian_Latin_100_CS_AS
GO

USE LogovanjeMVC
GO

CREATE TABLE Korisnik(
KorisnikId int NOT NULL IDENTITY(1,1) PRIMARY KEY,
KorisnickoIme nvarchar(50) NOT NULL UNIQUE,
Lozinka nvarchar(20) NOT NULL,
Provera UNIQUEIDENTIFIER DEFAULT NEWID()
);

INSERT INTO Korisnik(KorisnickoIme,Lozinka)
VALUES('student','student1');
```

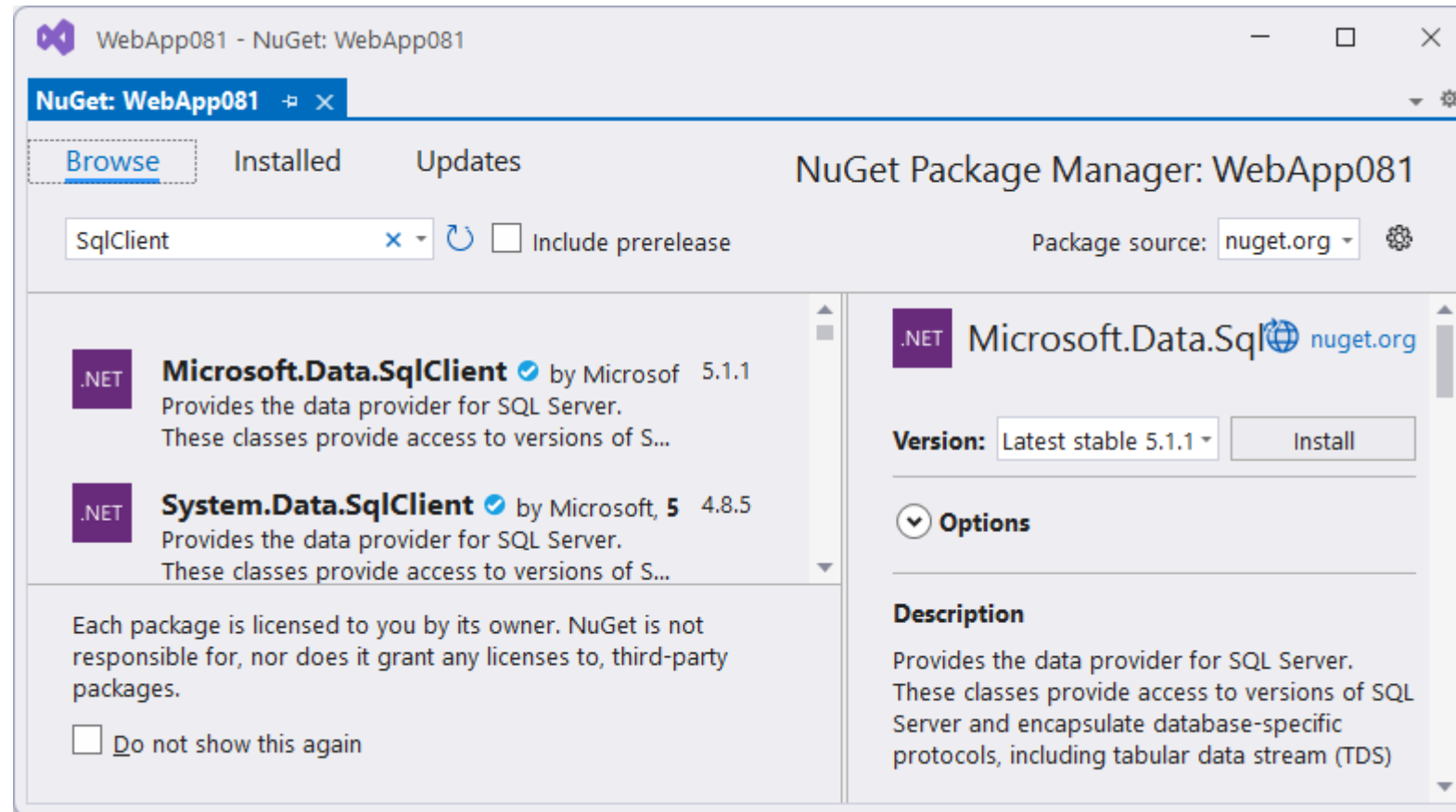
KorisnikId	KorisnickoIme	Lozinka	Provera
1	student	student1	AC761011-C027-48B7-A2F2-BF72EA959805

# Entitetska klasa Korisnik (Models)

```
public class Korisnik
{
    public int KorisnikId { get; set; }
    public string KorisnickoIme { get; set; }
    public string Lozinka { get; set; }

    public Guid Provera { get; set; }
}
```

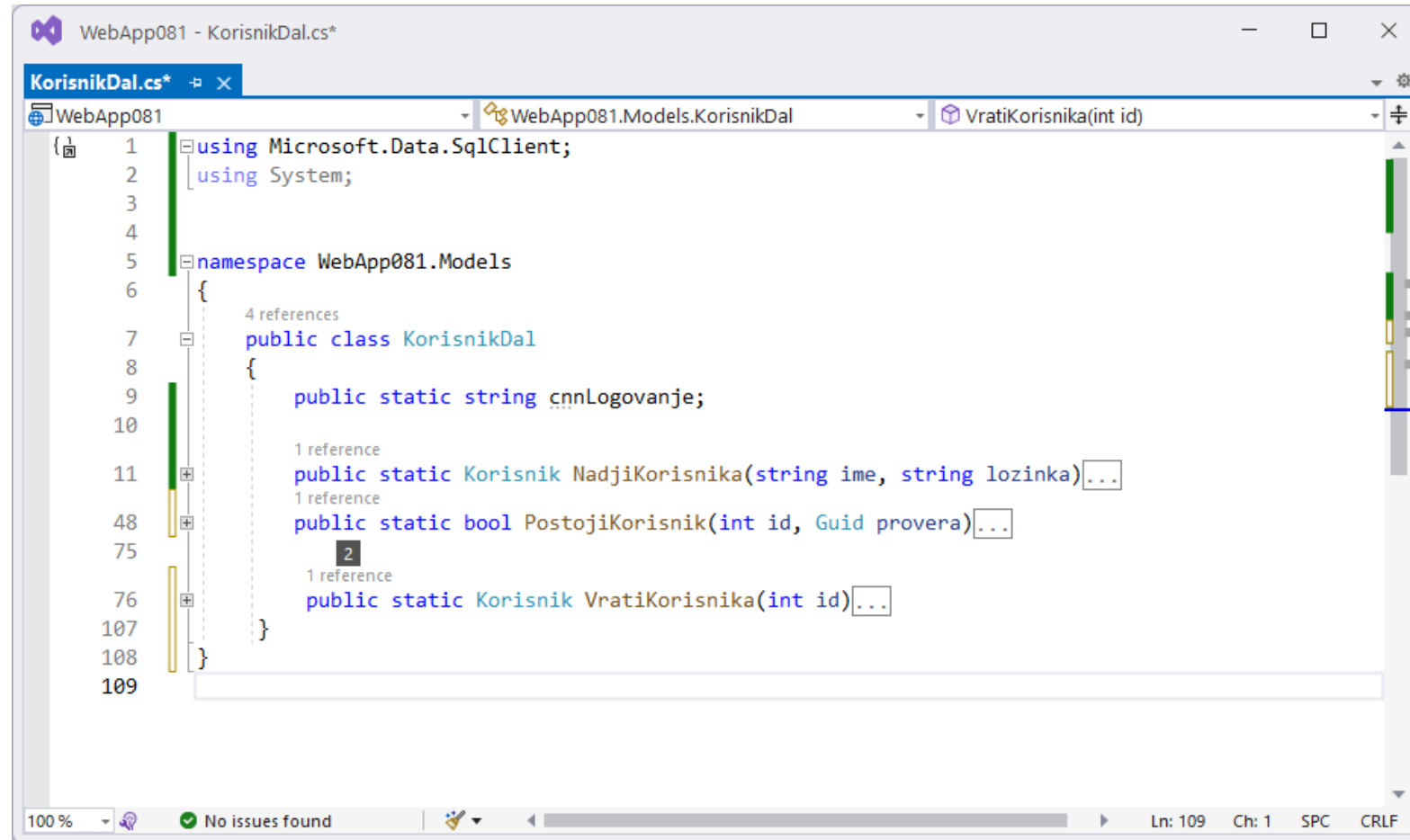
# Biblioteka Microsoft.Data.SqlClient



# appsettings.json

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "DefaultConnection": "Server=GORAN-HP;Initial Catalog=LogovanjeMVC;Integrated
Security=true; Encrypt=False"
  }
}
```

# Klasa KorisnikDal(models)



```
WebApp081 - KorisnikDal.cs*
KorisnikDal.cs*
WebApp081
WebApp081.Models.KorisnikDal
VratiKorisnika(int id)

1 using Microsoft.Data.SqlClient;
2 using System;
3
4
5 namespace WebApp081.Models
6 {
7     4 references
8     public class KorisnikDal
9     {
10         public static string cnnLogovanje;
11
12         1 reference
13         public static Korisnik NadjiKorisnika(string ime, string lozinka)...
14
15         1 reference
16         public static bool PostojiKorisnik(int id, Guid provera)...
17
18         2
19         1 reference
20         public static Korisnik VratiKorisnika(int id)...
21
22     }
23 }

100% No issues found Ln: 109 Ch: 1 SPC CRLF
```

```

public static Korisnik NadjiKorisnika(string ime, string lozinka)
{
    string upit = @"SELECT * FROM Korisnik
        WHERE KorisnickoIme = @ime AND Lozinka = @lozinka
        COLLATE Serbian_Latin_100_CS_AS";
    using (SqlConnection konekcija = new SqlConnection(cnnLogovanje))
    {
        SqlCommand komanda = new SqlCommand(upit, konekcija);
        komanda.Parameters.AddWithValue("@ime", ime);
        komanda.Parameters.AddWithValue("@lozinka", lozinka);
        try
        {
            konekcija.Open();
            SqlDataReader dr = komanda.ExecuteReader();
            if (dr.HasRows)
            {
                dr.Read();

                Korisnik k1 = new()
                {
                    KorisnikId = dr.GetInt32(0),
                    KorisnickoIme = dr.GetString(1),
                    Lozinka = dr.GetString(2),
                    Provera = dr.GetGuid(3)
                };
                return k1;
            }

            return null;
        }
        catch
        {
            return null;
        }
    }
}

```



```

public static bool PostojiKorisnik(int id, Guid provera)
{
    string upit = @"SELECT COUNT(*) FROM Korisnik
        WHERE KorisnikId = @KorisnikId AND Provera = @Provera";

    using (SqlConnection konekcija = new SqlConnection(cnnLogovanje))
    {
        SqlCommand komanda = new SqlCommand(upit, konekcija);
        komanda.Parameters.AddWithValue("@KorisnikId", id);
        komanda.Parameters.AddWithValue("@Provera", provera);
        try
        {
            konekcija.Open();
            int postojiKorisnik = (int)komanda.ExecuteScalar();
            if (postojiKorisnik == 1)
            {
                return true;
            }
            return false;
        }
        catch
        {
            return false;
        }
    }
}

```

```

public static Korisnik VratiKorisnika(int id)
{
    string upit = @"SELECT * FROM Korisnik WHERE KorisnikId = @KorisnikId";
    using (SqlConnection konekcija = new SqlConnection(cnnLogovanje))
    {
        SqlCommand komanda = new SqlCommand(upit, konekcija);
        komanda.Parameters.AddWithValue("@KorisnikId", id);
        try
        {
            konekcija.Open();
            SqlDataReader dr = komanda.ExecuteReader();
            if (dr.HasRows)
            {
                dr.Read();
                Korisnik k1 = new Korisnik
                {
                    KorisnikId = dr.GetInt32(0),
                    KorisnickoIme = dr.GetString(1),
                    Lozinka = dr.GetString(2),
                    Provera = dr.GetGuid(3)
                };
                return k1;
            }
            return null;
        }
        catch
        {
            return null;
        }
    }
}

```

# Kontroler za autentifikaciju

```
public class AccountController : Controller
{
    private readonly string _cnnLogovanje;

    public AccountController(IConfiguration configuration)
    {
        _cnnLogovanje = configuration.GetConnectionString("DefaultConnection");
        KorisnikDal.cnnLogovanje = _cnnLogovanje;
    }
}
```

# Svojstvo HttpContext

- Može mu se pristupiti unutar akcione metode kontrolera
- Vraća HttpContext objekat koji sadrži sve informacije o jednom HTTP zahtevu
- Svojstvo HttpContext.**User** daje informaciju o korisniku koji upućuje zahtev
- Svojstvo **Request** ovog objekta daje HttpRequest objekat zahteva i sadrži informacije o korisničkom zahtevu
- Svojstvo **Response** daje HttpResponse objekat za dati zahtev i sadrži informacije koje server šalje klijentu kao odgovor na zahtev
- Svojsva Request i Response mogu se pozivati bez prethodno pozivanja svojstva HttpContext
- Svojstvo HttpContext.**Session** vraća objekat koji omogućava rad sa korisničkom sesijom

# Metoda KreirajKolacic AccountController

```
using Microsoft.AspNetCore.Http;
```

```
private void KreirajKolacic(string kljuc, string vrednost, int? vremeVazenja = null)
{
    CookieOptions opcije = new CookieOptions();

    opcije.IsEssential = true;
    opcije.HttpOnly = true;

    if (vremeVazenja.HasValue)
    {
        opcije.Expires = DateTime.Now.AddMinutes(vremeVazenja.Value);
    }
    Response.Cookies.Append(kljuc, vrednost, opcije);
}
```

# Metoda ObrisiKolacic AccountController

```
private void ObrisiKolacic(string kljuc)
{
    Response.Cookies.Delete(kljuc);
}
```

# Akcija Login (GET) kontroler Account

```
public IActionResult Login(int id = 0)
{
    ViewBag.id = id;
    return View();
}
```

# Login.cshtml

```
<div class="text-center">
  <form action="/Account/Login" method="post">
    <label for="TextIme">Unesite korisnicko ime</label> <br>
    <input type="text" name="ime" id="TextIme"> <br>

    <label for="TextLozinka">Unesite lozinku</label> <br>
    <input type="password" name="lozinka" id="TextLozinka"> <br>
    <br>
    <input type="submit" value="Prijava">
  </form>
  <div>
    @if (ViewBag.id == 1)
    {
      <span>Neuspesno logovanje</span>
    }
    @if (ViewBag.id == 2)
    {
      <span>Morate se prijaviti na sistem</span>
    }
    @if (ViewBag.id == 3)
    {
      <span>Odjavljeni ste</span>
    }
  </div>
</div>
```



# Metoda Login POST

```
[HttpPost]
public IActionResult Login(string ime, string lozinka)
{
    Korisnik k = KorisnikDal.Nadjikorisnika(ime, lozinka);
    if (k != null)
    {
        KreirajKolicic("id", k.KorisnikId.ToString());
        KreirajKolicic("provera", k.Provera.ToString());

        return RedirectToAction("Index", "Home");
    }
    else
    {
        return RedirectToAction("Login", new { id = 1 });
    }
}
```

# Metoda Odjava kontroler Account

```
[HttpPost]
    public ActionResult Odjava()
    {
        Obrisikolacic("id");
        Obrisikolacic("provera");
        return RedirectToAction("Login", new { id = 3 });
    }
```

# Metoda Autentifikacija HomeController

```
private bool Autentifikacija()
{
    string id1 = Request.Cookies["id"];
    string provera1 = Request.Cookies["provera"];

    if (id1 != null && provera1 != null)
    {
        Guid provera = Guid.Parse(provera1);
        int id = int.Parse(id1);
        return KorisnikDal.PostojiKorisnik(id, provera);
    }
    return false;
}
```

# Metoda Index kontroler Home

```
public IActionResult Index()
{
    if (!Autentifikacija())
    {
        return RedirectToAction("Login", "Account", new { id = 2 });
    }
    int id = int.Parse(Request.Cookies["id"]);
    Korisnik k1 = KorisnikDal.VratiKorisnika(id);
    return View(k1);
}
```

# Pogled Index.cshtml

```
@model Korisnik
@{
    ViewData["Title"] = "Pocetna";
}

<h1>Dobro dosli @Model.KorisnickoIme</h1>
<br />
<h2>Vidite vas racun</h2>
<a href="/home/racun">Racun</a>

<form action="/Account/Odjava" method="post">
    <button>Odjava</button>
</form>
```

# Metoda Racun kontroler Home

```
public IActionResult Racun()
{
    if (!Autentifikacija())
    {
        return RedirectToAction("Login", "Account", new { id = 2 });
    }
    return View();
}
```

# Pogled Racun.cshtml

```
Vase stanje je: .....<br />
```

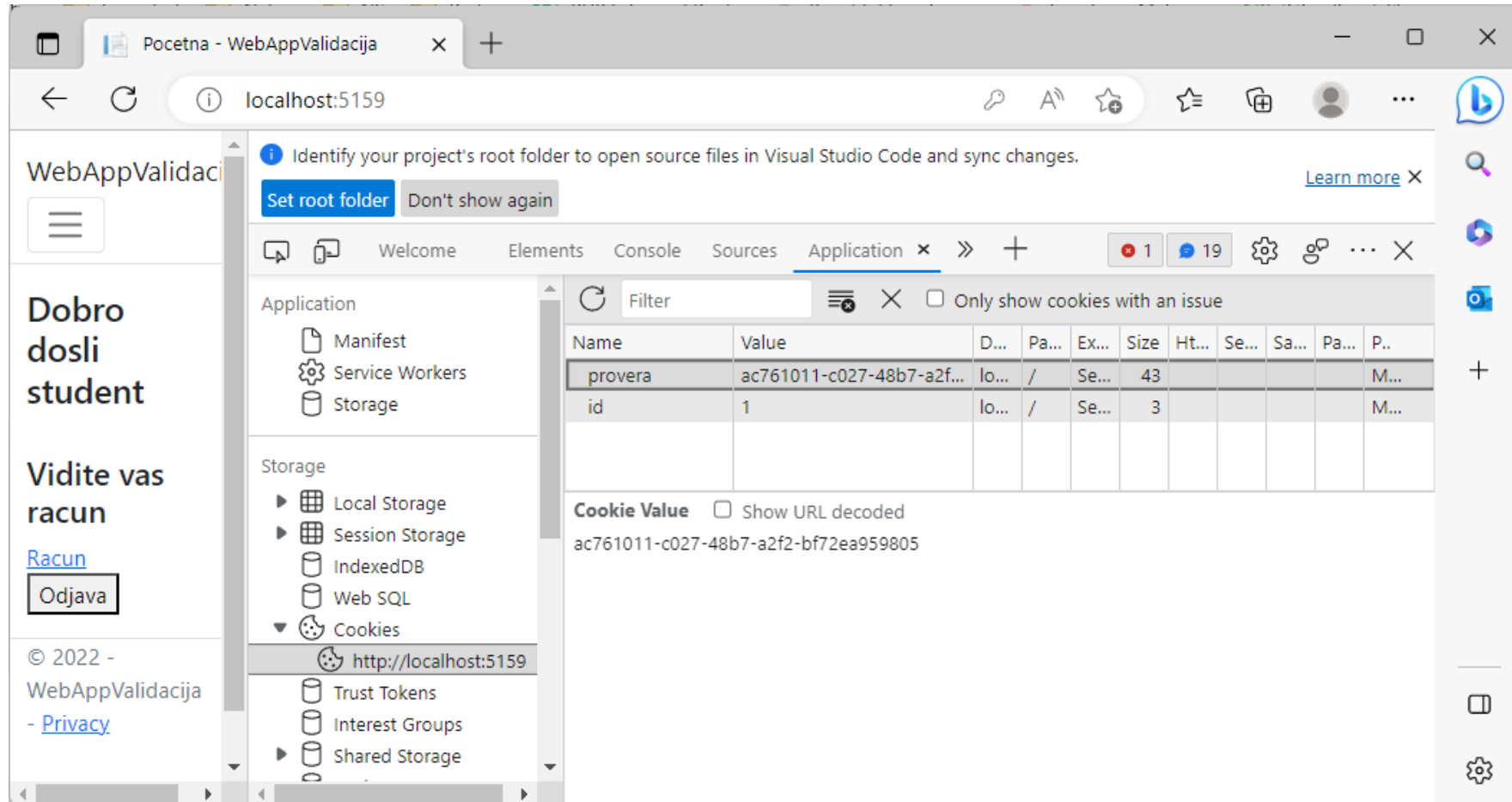
```
<a href="/home/index">Index</a>
```

```
<form action="/Account/Odjava" method="post">
```

```
  <button>Odjava</button>
```

```
</form>
```

# Prikaz kolačića u browseru



The screenshot shows a web browser window with the address bar set to `localhost:5159`. The page content includes a sidebar with a hamburger menu, a greeting "Dobro dosli student", and a "Vidite vas racun" section with a "Racun" link and an "Odjava" button. The footer contains copyright information for 2022 and a "Privacy" link.

The browser's developer tools are open to the "Application" tab. The left sidebar of the developer tools shows the "Storage" section expanded to "Cookies". A table of cookies is displayed with the following data:

Name	Value	D...	Pa...	Ex...	Size	Ht...	Se...	Sa...	Pa...	P..
provera	ac761011-c027-48b7-a2f...	lo...	/	Se...	43					M...
id	1	lo...	/	Se...	3					M...

Below the table, the "Cookie Value" section is expanded, showing the value `ac761011-c027-48b7-a2f2-bf72ea959805` for the selected cookie.



Sesija

# Sesija u ASP.NET Core web aplikacijama

- Predstavlja korisnika koji posećuje web aplikaciju
- Sesija započinje kada korisnik zatraži neku od serverskih metoda
- Sesija iščezava kada protekne određeni period vremena (koji se može konfigurirati korišćenjem svojstva `IdleTimeout`) od poslednjeg zahteva
- ASP.NET Core dodeljuje korisniku kolačić koji sadrži Id sesije
- Sesija koja koristi kolačiće ne alocira prostor za sesiju sve dok se podaci ne sačuvaju unutar sesionog objekta
- Podrazumevano se sesione vrednosti čuvaju u memoriji servera
- Sesija omogućava deljenje vrednosti između zahteva i čuva informacije specifične za konkretnog korisnika
- Stanje sesije se implementira kao kolekcija name, value parova

# Konfigurisanje sesije

- Da bi se omogućio middleware za sesiju potrebno je u fajlu Program.cs izvršiti:
- Implementaciju skladišta za sesiju( sesija se čuva kao objekat na serveru) – **AddDistributedMemoryCache**
- Pozvati metodu **AddSession** čime se registruje servis za sesiju
- Pozvati metodu **UseSession**

# Konfigurisanje sesije

```
builder.Services.AddDistributedMemoryCache();
//builder.Services.AddSession();
builder.Services.AddSession(options =>
{
    options.IdleTimeout = TimeSpan.FromMinutes(10);
    options.Cookie.HttpOnly = true;
    options.Cookie.IsEssential = true;
    options.Cookie.Name = "Login";
});

var app = builder.Build();

// Configure the HTTP request pipeline.
if (!app.Environment.IsDevelopment())
{
    app.UseExceptionHandler("/Home/Error");
}
app.UseStaticFiles();
app.UseSession();

app.UseRouting();

app.UseAuthorization();
```

# Upis podataka u sesiju

- Unutar kontrolera sesiji se pristupa pomoću svojstva `HttpContext.Session`
- `HttpContext.Session.Id` vraća broj sesije dodeljene korisniku
- Metoda `HttpContext.Session.SetString("key","value")` upisuje string vrednost u sesiju
- Metoda `HttpContext.Session.SetInt32("key",value)` upisuje celobrojnu vrednost u sesiju
- Metoda `HttpContext.Session.GetString("key")` vraća string vrednost sačuvanu u sesiji
- Metoda `HttpContext.Session.GetInt32("key")` vraća tip `int`? koja sadrži celobrojnu vrednost sačuvanu u sesiji
- Metoda `Clear()` briše sadržaj sesije

# Id sesije

```
using Microsoft.AspNetCore.Http;
```

```
public IActionResult Index()
{
    if (HttpContext.Session.GetString("brojSesije") == null)
    {
        HttpContext.Session.SetString("brojSesije", HttpContext.Session.Id);
    }

    ViewBag.brojSesije = HttpContext.Session.GetString("brojSesije");
    return View();
}
```

# Index.cshtml

```
@{  
    ViewData["Title"] = "Home Page";  
}  
  
<div class="text-center">  
    <span>Sesija: @ViewBag.brojSesije</span>  
    <br />  
    <a href="/Home/Strana2">Strana2</a>  
</div>
```

# Metoda Strana2

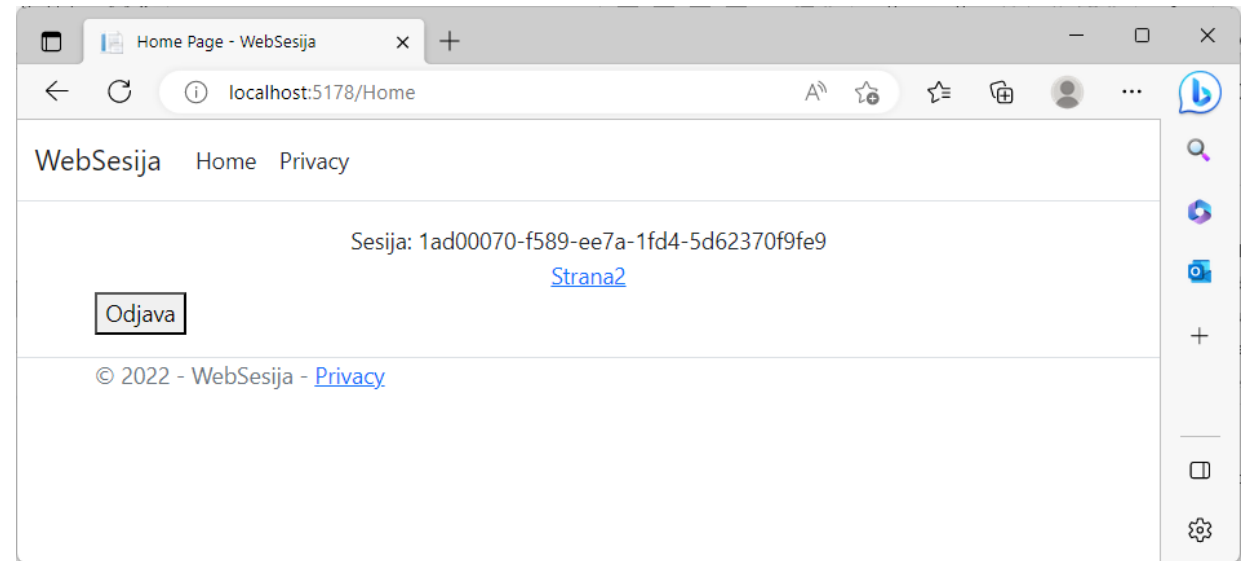
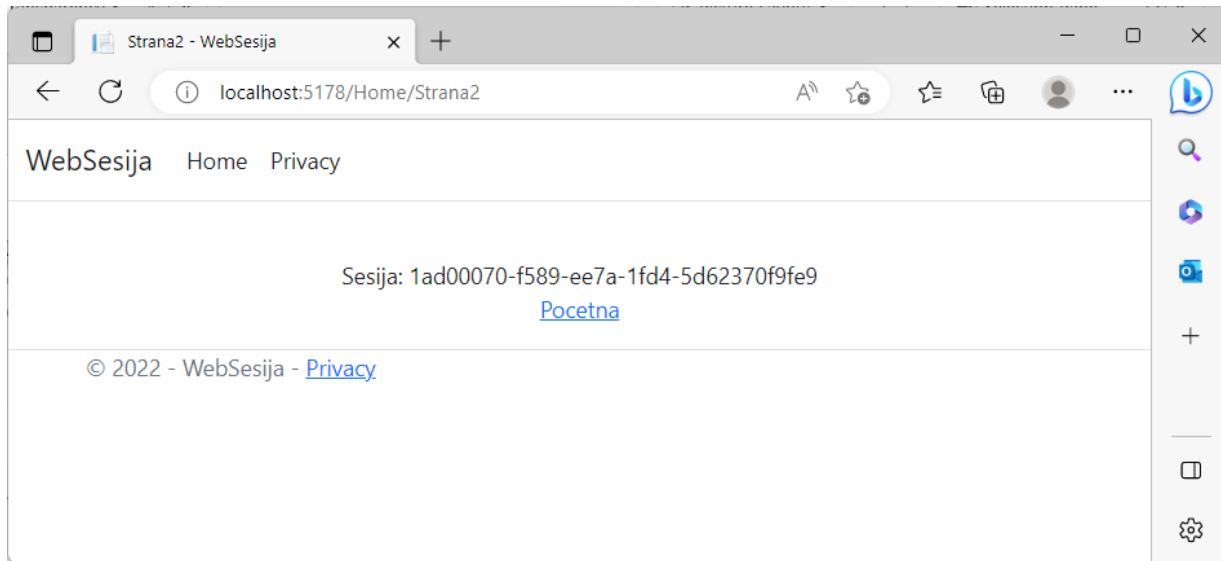
```
public IActionResult Strana2()
{
    ViewBag.brojSesije = HttpContext.Session.GetString("brojSesije");
    return View();
}
```



# Pogled Strana2.cshtml

```
@{  
    ViewData["Title"] = "Strana2";  
}  
  
<div class="text-center">  
    <span>Sesija: @ViewBag.brojSesije</span>  
    <br />  
    <a href="/Home">Pocetna</a>  
</div>
```

# Sesija



# Kolačić sa informacijama o korisničkoj sesiji (sesioni kolačić)

The screenshot shows a web browser window with the URL `localhost:5178/Home/Strana2`. The browser's developer tools are open, displaying the 'Application' tab. The 'Storage' section is expanded, showing 'Local Storage', 'Session Storage', and 'IndexedDB'. The 'Session Storage' section is further expanded, showing a single session cookie with the following details:

Name	Value	D...	P...	E...	S...	Http...	S	Sa...	P.	P.
Login	CfDJ8IXkRQSWb91...	l...	/	S...	1...	✓		Lax		M..

The 'Cookie Value' section is also expanded, showing the decoded value of the cookie: `CfDJ8IXkRQSWb91NvnaXzX4PxXAVTdg1VS54mqhkQy3NcTa%2BKwh8UpRVKmbXE u3Fj5zEL4PnChYmi0%2FrnWN%2BISw8uWN39b4hAScwq2NqCYG%2BbE7yoFwzV% 2B4q12dtm36DAHo1KwRRooh04vNxxvHUmdbLR%2Fg7z1yh935bll0Z3wCPIHyJ`.

# AccountController

```
[HttpPost]
public IActionResult Login(string ime, string lozinka)
{
    Korisnik k = KorisnikDal.Nadjikorisnika(ime, lozinka);
    if (k != null)
    {
        HttpContext.Session.SetInt32("id", k.KorisnikId);
        HttpContext.Session.SetString("provera", k.Provera.ToString());

        return RedirectToAction("Index", "Home");
    }
    else
    {
        return RedirectToAction("Login", new { id = 1 });
    }
}
```

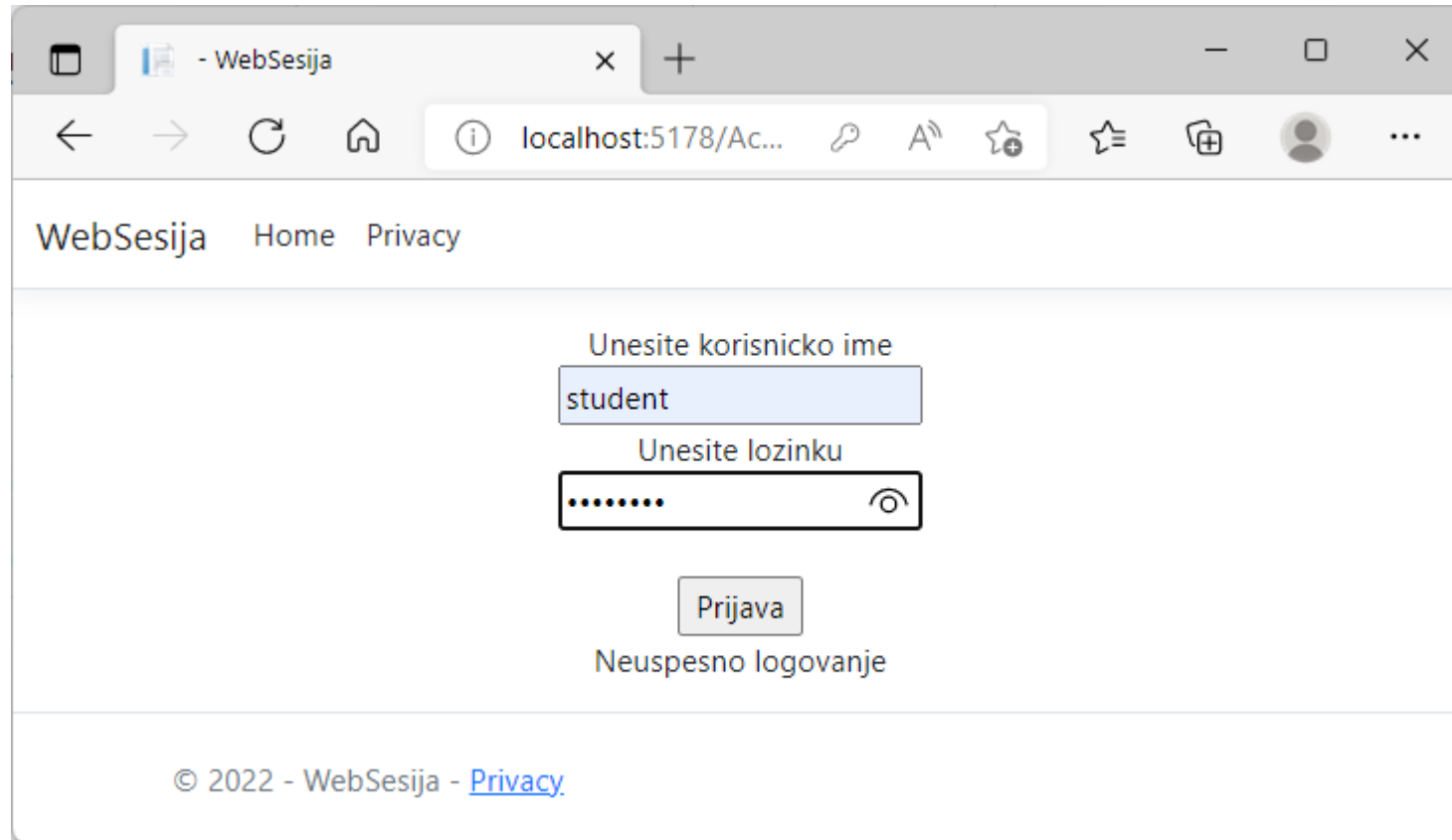
Metoda za logovanje uz kreiranje sesije

# AccountController

```
[HttpPost]
public ActionResult Odjava()
{
    HttpContext.Session.Clear();
    return RedirectToAction("Login", new { id = 3 });
}
```

Metoda za odjavljivanje

# Strana za logovanje



The image shows a web browser window with a single tab titled '- WebSesija'. The address bar displays 'localhost:5178/Ac...'. The page content includes a navigation menu with 'WebSesija', 'Home', and 'Privacy'. The main form area contains the following elements:

- Label: 'Unesite korisnicko ime' (Enter username)
- Input field: 'student'
- Label: 'Unesite lozinku' (Enter password)
- Input field: masked with dots and a visibility toggle icon
- Button: 'Prijava' (Login)
- Message: 'Neuspesno logovanje' (Failed login)

At the bottom of the page, there is a footer: '© 2022 - WebSesija - [Privacy](#)'.

# Sesioni kolačić

The screenshot shows a web browser window with the Application tab open. The left sidebar contains a navigation menu for 'WebSesija' with a hamburger menu icon, and a main content area with the text 'Dobro dosli student' and 'Vidite vas racun'. Below this, there is a link 'Racun' and a button 'Odjava'. The footer of the page reads '© 2022 - WebSesija - E'.

The Application tab is active, showing a tree view of storage areas on the left. The 'Cookies' folder is expanded, showing a sub-entry for 'http://localhost:5178'. The main pane displays a table of cookies for this domain.

Name	Val...	Domain	Path	Expires / Max-Age	S...	H	Sec...	Sa...	Sa...	Par...	P...
.AspNetCore.Session	Cf...	localhost	/	Session	2...	✓		Lax			Me...
ssupp.visits	11	localhost	/	2022-09-20T13:09...	14			Strict			Me...
ssupp.vid	vifl...	localhost	/	2022-09-20T13:09...	22			Strict			Me...

Below the table, the 'Cookie Value' section is visible, with a checkbox for 'Show URL decoded'. The value is a long alphanumeric string: CfDJ8Lkles1bCzFBqkAAeNNY2Z8%2F3zqzlxNHhbl4KaLf8763iGmSVBv4aVQV5MJawQaK2tJRqZQdyO21hAShAEkRQ%2BhTdtCRmNlaEDUz09oBZFn3SaWAs8FBVpTSWXD3fwyZaKsKI%2FhLHz5LhGcBGGtjPo6%2ByAg8EftiAJinnL%2FnnUUy

# HomeController

```
private bool Autentifikacija()
{
    int? id1 = HttpContext.Session.GetInt32("id");
    string provera1 = HttpContext.Session.GetString("provera");

    if (id1 != null && provera1 != null)
    {
        return true;
    }
    return false;
}
```

Provera postojanja sesionih promenljivih



# HomeController

```
public IActionResult Index()
{
    if (!Autentifikacija())
    {
        return RedirectToAction("Login", "Account", new { id = 2 });
    }
    int id = HttpContext.Session.GetInt32("id").Value;
    Korisnik k1 = KorisnikDal.VratiKorisnika(id);
    return View(k1);
}
```

# Index.cshtml

```
@model Korisnik

<h1>Dobro dosli @Model.KorisnickoIme</h1>
<br />
<h2>Vidite vas racun</h2>
<a href="/home/strana2">Strana2</a>

<form action="/Account/Odjava" method="post">
  <button>Odjava</button>
</form>
```

# Pitanje 1

Da bi smo pristupili sesionom objektu unutar metode kontrolera pišemo:

- a. `Cookie.Session`
- b. `HttpContext.Session`
- c. `Session`

Odgovor: b

# Pitanje 3

Da bi se zabranilo da klijentski kod može da iščitava kolačić potrebno je postaviti sledeće svojstvo CookieOptions objekta:

- a. HttpOnly
- b. ServerOnly
- c. Disabled

Odgovor: a

# Pitanje 3

Upis string vrednosti u sesioni objekat ispravno je izvršen u sledećem primeru:

- a. `Session.Set("ime","value")`
- b. `HttpContext.Session.SetString(ime,"value")`
- c. `HttpContext.Session.SetString("ime","value")`

Odgovor: c

# Pitanje 4

Čitanje tekstualne vrednosti koja je upisana u sesiju pod ključem ime vrši se na sledeći način:

- a. `string ime = HttpContext.Session.GetString(ime);`
- b. `string ime = HttpContext.Session.GetString("ime");`
- c. `string ime = HttpContext.GetString("ime");`

Odgovor: b